

### **Remarks**

Applicants respectfully request reconsideration of the present application in view of the foregoing amendments and the following remarks. Claims 1-17 are pending in the application. Claims 1-17 are rejected. No claims have been allowed. Claims 1, 10, and 13 are independent. Editorial amendments have been made to claims 1, 3, 10, 13, and 17. No new matter is added.

### ***Cited Art***

The Action cites:

Kim et al., "Design and Implementation of Home Network Systems Using UPnP Middleware for Networked Appliances," IEEE Transactions on Consumer Electronics, Volume 48, Issue 4, Nov 2002, pages 963-972 ("Kim");

Kumar et al (US Patent No. 7,017,148) ("Kumar");

Skingsley et al. (US Patent No. 6,697,751) ("Skingsley"); and

UPnP Implementers Corporation, "UPnP Device Certification Process Document," Version 1.0, 2001) ("UPnPIC").

### ***Claim Objections***

The Action objects to the language used in claim 1, suggesting that it would be better if written without the word "to" before the words "which action." [See, Action, at § 5, pages 2-3.] Applicants thank the Examiner for the suggestion and have so amended the claim.

The Action also objects to the use of the term "defect configuration" in claims 10 and 17, alleging that "the claim language could be confusing because it is unclear what the 'defect configuration' is a 'defect configuration' of." [Action, at § 6, page 3.] While Applicants respectfully disagree that the language is confusing, in the interest of expediting examination Applicants have amended claims 10 and 17 to recite "device defect configuration file"(s). With these amendments, and for the reasons discussed below, Applicants respectfully submit that the claims are in condition for allowance.

### ***Claim Rejections - 35 USC § 112***

The Action rejects Claims 1-12 under 35 USC § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

In particular, the Action rejects claims 1-12 as being directed to emulating devices “in a device connectivity protocol.” The rejection states that “it is unclear how a device can be emulated in a ‘protocol.’” [Action, at § 10, pages 3-4.] While Applicants respectfully disagree that the claim language is indefinite, Applicants have amended independent claims 1 and 10 to recite “emulating devices *communicating through* a device connectivity protocol.”

The Action also rejects claim 3, stating that it is “unclear” whether the claim’s recited “action” is the action recited in claim 1. [Action, at § 10, page 4.] Applicants have since amended the claim to recite “the validated action” which Applicants note corresponds to the language “to validate which action out of the set of actions specified in the description the action request matches” as recited in claim 1.

The Action also rejects claim 3, stating that there is insufficient antecedent basis for the language “the evented variables” previously found in the claim. [See, Action, at § 10, page 4.] The claim has since been amended to remove the word “the” from the phrase, and thus should no longer provide an antecedent basis problem.

Applicants believe that, with these amendments, the claims 1-12 are not indefinite and thus are in condition for allowance.

### ***Claim Rejections under 35 USC § 102***

The Action rejects claims 1-3, 13, 15, and 16 under 35 USC 102(a) as being anticipated by Kim. For a 102(a) rejection to be proper, the cited art must show each and every element as set forth in a claim. (See MPEP § 2131.01.) However, the cited art does not describe each and every element. Accordingly, applicants request that the rejection be withdrawn. Claims 1 and 13 are independent.

*Claim 1*

Claim 1, as amended, recites:

processing, in a *generic device emulator able to emulate more than one device based on device descriptions*, a description of a device to be emulated in the device connectivity protocol, the description specifying a set of actions of the device to be emulated;

in response to receiving an action request at the device emulator per the device connectivity protocol, *checking the action request against the device description in the device emulator to validate which action out of the set of actions specified in the description the action request matches;*

[Emphasis added.] For example, the Application, describes the utility of generic device emulation:

In emulating a device based on its description, the generic device emulator provides default behaviors for a set of capabilities defined in the description (e.g., for a set of actions defined in UPnP™ service descriptions for services of the device). . . . *The device developer therefore need not provide specific behavior implementations of the device's capabilities, and the generic device emulator still provides an emulation of the device operating within the protocol meeting the device's definition. Further, the device developer can provide specific behavior implementations of none, some or all of the device's capabilities, as desired. . . . The generic device emulator supplies the default behavior of those capabilities for which no specific behavior implementation is provided.*

[Application, at page 4, lines 11-25; emphasis added.] The Application then goes on to describe more specific examples of device emulation, including the checking of an action against a description:

With reference now to Figure 2, *the generic device emulator 210 emulates the operation of an emulated device (e.g., devices 130-132) within the network architecture 100 (Figure 1) of UPnP™, including the behavior of the emulated device during Addressing, Discovery, Description, Control and Eventing phases of UPnP™. In other words, the generic device emulator effectively provides an implementation of the emulated device as it would operate within the UPnP™ protocol. . . .*

The generic device emulator 210 emulates the behaviors of the emulated device within UPnP™ based on the UPnP™ description of the device . . . . Given the device and service descriptions 220-221 of any UPnP™ device, the generic device emulator 210 emulates behaviors implementing the description within UPnP™. . . .

In the control phase, the generic device emulator 210 implements default behaviors for the actions specified in the emulated device's service description document(s) 221. *The generic device emulator 210 receives action invocation messages (SOAP commands) from control points and validates these messages against the emulated device's service description.* Upon validation, the generic

device emulator 210 provides a default response to the action invocation, which response conforms to the data format and types specified in the service description for the action. For example, if the response to the action specified in the service description is to return an integer value, the default behavior simply returns a default integer value (e.g., a zero).

[Application, at page 13, line 2, to page 14, line 2.] The application goes on to discuss the acquisition of a device description at page 16:

Device and Service Info

Once a device description document 220 is given to the emulator 210, the device info and all the service info are parsed and are maintained by the Device and Service Info object 320. The object 320 provides methods by which the other components of the emulator can get information about the device and its services.

[Application, at page 16, lines 20-24.] At page 15, the Application makes clear the benefits of parsing and checking the UPnP descriptions in device emulation:

*The generic device emulator 210 thus provides emulation of any device in the UPnP™ protocol, based on no more than the UPnP™ description of the device.* A vendor can therefore produce an implementation of a UPnP™ description to propose for standardization, with no further work than defining the UPnP™ description itself and without having to individually implement the various protocols involved in UPnP™ (e.g., HTTP, SOAP and SSDP) for the specific device. Further, the vendor can easily add implementations of specific actions to over-ride default behaviors, such as to provide state-dependent action responses. Further, the vendor can provide defect filters to introduce defective behaviors, such as for testing purposes.

[Application, at page 15, lines 5-13; emphasis added.]

*Kim's "appliance emulators" do not check against a description to operate since each only uses descriptions to advertise their operations. Therefore they do not teach or suggest "processing, in a generic device emulator able to emulate more than one device based on device descriptions," or "checking the action request against the device description in the device emulator to validate which action" as recited in claim 1.* Applicants note first that, while Kim illustrates examples of XML descriptions of air-conditioner and toaster emulators at page 969, Kim does not describe the processing of these descriptions by the appliance emulators themselves. Instead, this XML description is used only to *advertise* to Kim's home server information so the server can interact with the emulator. Thus, there is no indication in Kim that these descriptions are "processed," or that actions are "checked" against the description in order to "validate" the action by the appliance emulators of Kim.

In particular, the only passage of Kim describing the use of XML files reads:

When an air conditioner emulator is plugged into the network, it sends device information to advertise its existence. Then, *after the home server receives information about the home appliance using the XML page, the action service is activated.*

[Kim, at page 968, column 1, paragraph 3.] Through this passage, Kim describes the *sending* of an XML description from an appliance emulator to a home server. Thus, Kim does not show “processing” or “checking” *performed by a device emulator* as in claim 1. Instead, any processing is performed by the home server upon receiving the XML file. Applicants assume that the Action does not imply that the home server should read on the “device emulator” language of claim 1, since the Action, at § 56, page 18, notes that the citations used in the rejection are directed to Kim’s “appliance emulators” and not Kim’s “home server.” Hence, Applicants believe that Kim does not read on the above-quoted language of claim 1.

*Kim’s “appliance emulators” are each built for a specific device emulation job. Therefore they do not teach or suggest “processing, in a generic device emulator able to emulate more than one device based on device descriptions, a description of a device to be emulated” as recited claim 1.* As previously noted, Kim demonstrates examples of various appliance emulators at Table 2 and Figure 9. With regard to Table 2, Kim describes emulators based on different operating systems (e.g., the “O.S.” column) and with differing display types (e.g., the “display type” column). [Kim, at Table 2.] This demonstrates, at the least, that Kim’s appliance emulators were likely built separately, at least because they were built in different operating systems. This fact is supported by Figure 9, which shows four widely different user interfaces for four different appliance emulators.

These differences make it difficult for Kim to demonstrate generic emulation. It would be unlikely, if Kim appliance emulators were able to teach or suggest “processing, in a generic device emulator able to emulate more than one device based on device descriptions, a description of a device to be emulated” that different devices would be emulated on different platforms and with different GUIs. Indeed, Kim’s teaching of disparate emulations would teach away from the utility of using a device description.

The Action, in its Response to Arguments portion, notes that claim 1 previously did not recite “generic” device emulation outside of the preamble. [Action, at § 58, pages 19-20.] Applicants note that claim 1 has been amended to recite that the processing of the descriptions is

performed “in a generic device emulator able to emulate more than one device based on device descriptions.” Applicants believe the claim language, as amended, makes clear the distinction between Kim’s hard-wired appliance emulators and the generic device emulation of claim 1.

For at least these reasons, Kim cannot teach or suggest each and every element in claim 1. Thus, the rejection of claim 1 over Kim is improper. Applicants therefore request that the rejection of claim 1, and of dependent claims 2 and 3, be withdrawn and that claims 1-3 be allowed.

### *Claim 13*

the generic device emulator comprising:  
    program code for receiving action requests directed to the device  
    within the device connectivity architecture;  
    program code for checking an action requests against the  
    description to validate whether the action request matches that of an action  
    specified in the description; and  
    program code for performing a default behavior producing a  
    response for the action consistent with the data format specified in the  
    description, thereby emulating operation of the device for the action.

In its rejection of the quoted language in claim 13, the Action cites to similar portions of Kim as were discussed above with respect to claim 1. Thus, for at least the reasons discussed above with respect to claim 1, Kim cannot teach or suggest each and every element in claim 13. The rejection of claim 13 over Kim is improper. Applicants therefore request that the rejection of claim 13, and of dependent claims 15 and 16, be withdrawn and claims 13, 15, and 16 be allowed.

### ***Patentability of Claims 4-12, 14, and 17 Under 35 USC § 103(a)***

The Action rejects claims 4-12, 14, and 17 under 35 USC 103(a) as being unpatentable over a variety of references. In particular, the Action rejects independent claim 10 under 35 USC 103(a) as being unpatentable over Skingsley, in view of UPnPIC. To establish a prima facie case of obviousness, the prior art reference (or references when combined) must teach or suggest all the claim limitations. (MPEP § 2142.)

*Claim 10*

Claim 10, as amended, recites:

*reading a device defect configuration file representing in a tagged text format at least one defect behavior to be applied to a type of packet transmitted from an emulated device per the device connectivity protocol;*

*...*

*transmitting the packet from the emulated device as modified by applying the defect behavior*

[Emphasis added.] An example of using a device defect configuration file to direct the applications of defects is shown at page 14, line 27 to page 15, line 4:

The generic device emulator 210 further provides a mechanism (described more fully below) for the vendor to define defective behaviors, which can be useful for testing control points 110-111 (Figure 1). *The defective behaviors are defined using XML format statements in a textual configuration file, which describe defect filters to be applied to the messages generated by the generic device emulator for the emulated device.* For example, a defect filter can be defined to have the generic device emulator 210 strip off a leading ‘\*\*’ character from headers of SOAP messages sent for the emulated device.

[Emphasis added.]

*Skingsley applies changes to network transmissions on a random basis, and without reference to a device defect configuration file, and thus cannot teach or suggest the above-quoted language of claim 10.* In its rejection of claim 10, the Action cites to portions of columns 12 and 13 of Skingsley. These passages, however, show that Skingsley *does not* utilize a “device defect configuration file” when modifying network transmissions:

The aim of the emulator 701 is to simulate a variety of network conditions, for a variety of packets that embed a range of protocols, and over a range of types of networks. . . . *Using means 703 to effect a change to the transmission characteristics of the network shown in FIG. 7b, the emulator 701 subjects these packets to drop, delay, jitter, etc., which allows the associated applications to review their methods for handling such network interruptions.* . . .

*...*

Thus the means 703 to effect a change to the transmission characteristics of the network operates on the packets once they have been captured by the capturing means 705. The modified packet is then passed to the sending means 707 for injection back into the network (or not, depending upon predetermined rules). . . .

[Skingsley, at column 12, lines 31-63.] Skingsley’s lack of a device defect configuration file is made more clear in the portion of column 13 cited in the rejection, where Skingsley is shown to *apply changes randomly*:

The processes that affect network traffic are largely random in nature; thus, in order to realistically simulate network conditions, the packets that are interfered with by the above means are selected randomly by the emulator 701.

[Skingsley, at column 13, lines 63-66.] Thus, while Skingsley describes choosing the packets which are affected randomly, it does not at all discuss how it chooses *which* effects to apply to the packets to emulate network errors.

Applicants do not find further such disclosure in the other portions of Skingsley cited in the rejection. Applicants note that the cited passage of column 4 describes only the registration of protocol processes. This does not teach or suggest a “device defect configuration file” as recited in the claim, let alone the use of one. Similarly, cited Figures 4 and 7 show only that packets are handled (Figure 4) and that changes are applied to them (Figure 7). Neither goes into sufficient detail to teach or suggest this language of claim 10.

For at least these reasons, Skingsley does not teach or suggest the above-quoted language of claim 10. Furthermore, Applicants do not find relevant disclosure in UPnPIC. Thus, the rejection of claim 10 over Skingsley in view of UPnPIC is improper. Applicants request that the rejection of claim 10 be withdrawn. Applicants also request that the rejections of claims 11 and 12, each of which depend from claim 10, be withdrawn. Claims 10-12 should be allowed.

Finally, the Action rejects claims 4, 5, and 14 under 35 U.S.C § 103(a) as unpatentable over Kim in view of Kumar. The Action rejects claims 6, 7, and 9 under 35 U.S.C § 103(a) as unpatentable over Kim in view of Skingsley. The Action rejects claims 8 and 17 under 35 U.S.C § 103(a) as unpatentable over Kim in view of Skingsley and further in view of Kumar and UPnPIC.

Each of these claims depends from independent claims 1 and 13. Because Applicants do not find additional disclosure for the above-quoted language of claims 1 and 13 in either Kumar, Skingsley, or UPnPIC, Applicants respectfully submit that, for at least the reasons discussed above with respect to claims 1 and 13, the rejections of claims 4-9, 14, and 17 are improper. Applicants request that the rejections of claims 6-9, 14, and 17 be withdrawn and that the claims be allowed.



***Interview Request***

If the claims are not found by the Examiner to be allowable, the Examiner is requested to call the undersigned attorney to set up an interview to discuss this application.

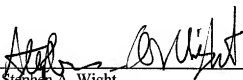
***Conclusion***

The claims in their present form should be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Telephone: (503) 595-5300  
Facsimile: (503) 595-5301

By   
Stephen A. Wight  
Registration No. 37,759